

Das CMake-Umfeld

Wolfgang Dautermann

15. Januar 2017

1 Motivation

Zum Programmieren braucht man nicht nur eine Programmiersprache und eine Idee sondern häufig auch ein Buildsystem, das notwendige Compiler, Zusatzpakete, etc. findet und das Gesamtprogramm compiliert.

Aber damit ist die Arbeit noch nicht fertig:

- Das Programm soll auch (automatisiert) getestet werden.
- Entwickler/innen sollen Build-Ergebnisse (beispielsweise von Nightly Builds) und Testergebnisse übersichtlich sehen.
- Und ausserdem soll das Programm auch noch paketiert werden, sodass es bequem über die Repositories der Linux-Distributionen installierbar ist. Um da möglichst schnell reinzukommen, kann man die Erstellung von Paketen selbst machen oder dem Package-Maintainer die Arbeit möglichst vereinfachen.
- Auch Installer für Windows sollen einfach erstellbar sein – am besten, ohne Windows anzugreifen...

2 Geplante Struktur des Vortrags

- Kurze Einführung in CMake
- Paketerstellung mit CPack: Neben Paketen für diverse Linux-Varianten werden – mittels Crosscompiling – auch Installer für Windows mit CMake/CPack erzeugt
- Programmtests mit CTest
- webbasierte Darstellung der Build- und Testergebnisse mit CDash

3 Zielgruppe

Entwickler/innen, die am Buildsystem CMake interessiert sind und an automatischer Paketierung ihrer Programme interessiert sind.

Die gezeigten Beispiele werden überwiegend in C sein, die Vorgehensweise ist auch auf andere (compilierbare) Programmiersprachen wie C++ oder Fortran übertragbar.

4 Praktische Vorführung

Zu den vorgestellten Programme und Methoden wird es selbstverständlich auch kurzen praktischen Vorführungen geben.

5 Über den Vortragenden

Der Vortragende erstellt – mittels Crosscompiling und umfangreichen CMake-Skripten – Installer für das Computeralgebrasystem Maxima, und – auch mittels CMake – nightly Builds für wxMaxima.

6 Literatur/Weblinks

- <http://www.cmake.org/>
- <http://www.cdash.org/>